



Fernando, WAC., Canagarajah, CN., & Bull, D. (2000). *Video special effects editing in MPEG-2 compressed video*. 281 - 284.
<https://doi.org/10.1109/ISCAS.2000.856316>

Peer reviewed version

Link to published version (if available):
[10.1109/ISCAS.2000.856316](https://doi.org/10.1109/ISCAS.2000.856316)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

VIDEO SPECIAL EFFECTS EDITING IN MPEG-2 COMPRESSED VIDEO

W.A.C. Fernando, C.N. Canagarajah, D. R. Bull

Image Communications Group, Centre for Communications Research, University of Bristol,
 Merchant Ventures Building, Woodland Road, Bristol BS8 1UB, United Kingdom.
 Voice - +44-117-954-5198, Fax - +44-117-954-5206
 Email - W.A.C.Fernando@bristol.ac.uk

ABSTRACT

With the increase of digital technology in video production, several types of complex video special effects editing have begun to appear in video clips. In this paper we consider fade-out and fade-in special effects editing in MPEG-2 [1] compressed video without full frame decompression and motion estimation. We estimated the DCT coefficients and use these coefficients together with the existing motion vectors to produce these special effects editing in compressed domain. Results show that both objective and subjective quality of the edited video in compressed domain closely follows the quality of the edited video in uncompressed video at the same bit rate.

1. INTRODUCTION

Video special effects are needed to enhance the quality of the video production. Most special effects can be divided into three major categories: dissolving, fading and wiping. All these special effects are used to produce a gradual scene change between two scenes. However, these video editing tools are designed for spatial domain processing. The large channel bandwidth and memory requirements for the transmission and storage of image and video necessitate the use of video compression techniques [1,2]. Hence, the visual data in multimedia databases is expected to be stored mostly in the compressed form. Thus video editing in compressed domain is also essential. Therefore, a typical desktop video editing system must first convert the compressed domain representation to a spatial domain representation and then perform the editing function on the spatial domain data. This increases the overall computational complexity of the editing process. In order to avoid the unnecessary decompression operations and compression processes, it is efficient to edit the image and video in the compressed format itself.

Smith et al [3] showed how the algebraic operation of pixel-wise and scalar addition and multiplication, can be done in DCT compressed domain. He used these operations in JPEG [4] images to implement two common video transformations: dissolving and sub-titling. Author argued that these scalar addition and multiplication could be implemented on quantised matrices. However due to the non-linear behaviour of the mapping function, many problems were introduced with this scheme [3]. Another limitation of this scheme is the problems associated with extending this scheme for video compression

standards such as MPEG-2. Shen proposed DC-only fade-out operation for MPEG compressed video [5]. This algorithm is proposed under the crude assumption that fade-out is viewed as a reduction of picture brightness. However, this is a poor assumption of the actual fade-out operation in video production. In this paper we present a novel technique for fade-out and fade-in editing in compressed video without full frame decompression and re-compression.

Rest of the paper is organised as follows. In section 2 we present the mathematical model for fading and DCT coefficients extraction in compressed domain. Proposed scheme for video special effects editing is presented in section 3. Some experimental results are given in section 4. Finally, section 5 presents the conclusions and future work.

2. VIDEO EDITING IN COMPRESSED DOMAIN

2.1 Video Special Effects

In video editing and production, proportions of two or more picture signals are simply added together so that the two pictures appear to merge on the output screen. Very often this process is used to move on from picture A to picture B. In this case, the proportions of the two signals are such that as the contribution of picture A changes from 100% to zero, the contribution of picture B changes from zero to 100%. This is called dissolving. When picture A is a solid color, it is called as fade-in and when picture B is a solid colour, it is known as fade-out. Mathematically, fade-in and fade-out can be expressed as shown in Equations 1-2 respectively.

$$S_n(x, y) = \begin{cases} f_n(x, y) & 0 \leq n < L_1 \\ \left[1 - \left(\frac{n-L_1}{F}\right)\right]C + \left(\frac{n-L_1}{F}\right)g_n(x, y) & L_1 \leq n \leq (L_1 + F) \\ g_n(x, y) & (L_1 + F) < n \leq L_2 \end{cases} \quad (1)$$

$$S_n(x, y) = \begin{cases} f_n(x, y) & 0 \leq n < L_1 \\ \left[1 - \left(\frac{n-L_1}{F}\right)\right]f_n(x, y) + \left(\frac{n-L_1}{F}\right)C & L_1 \leq n \leq (L_1 + F) \\ g_n(x, y) & (L_1 + F) < n \leq L_2 \end{cases} \quad (2)$$

where, C is the video signal level (solid value), $S_n(i, j)$ is the resultant video signal, $f_n(i, j)$ is picture A, $g_n(i, j)$ is picture B, L_1 is length of sequence A, F is length of fading sequence, L_2 is length of the total sequence.

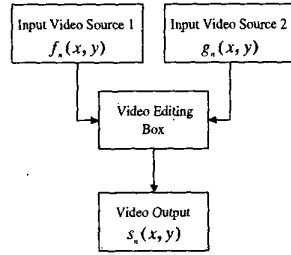


Figure 1: Uncompressed domain video editor

Figure 1 illustrates an uncompressed domain video editor. The function of the editing box is to scale input video signals and add these two signals to produce fading. When both $f(x, y)$ and $g(x, y)$ are compressed, this operation cannot be done in compressed domain without some additional processing. A straightforward way to performing fading on compressed sequence would be to decompress the sequence, apply fading operation in the spatial domain and recompress the video. Figure 2 presents this model. However the costly IDCT, DCT and motion estimation operations make it difficult for real time applications. Here, we propose an alternative scheme to do fading operation directly in compressed domain with minimum decompression.

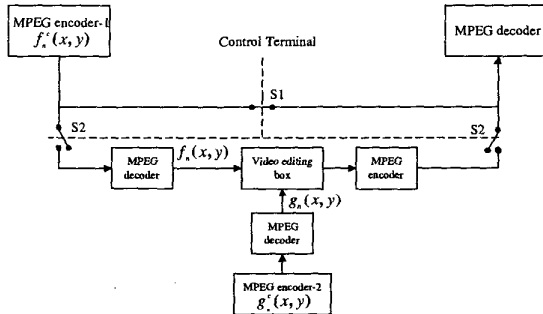


Figure 2: Conventional compressed domain video editor
(f_n^c - Compressed signal of f_n and g_n^c - Compressed signal of g_n)

2.2 DCT Coefficients Estimation

In this section we show how DCT coefficients are estimated for inter-coded frames in MPEG-2 video.

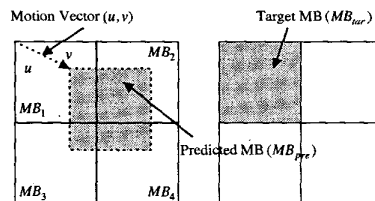


Figure 3: Graphical representation of four neighbouring MBs and motion vectors

$$DCT(MB_{pre}) = \sum_{i=1}^4 DCT(S_{i1}) DCT(MB_i) DCT(S_{i2}) \quad (3)$$

Sub-block	Position	S_{i1}	S_{i2}
MB_1	lower right	$\begin{bmatrix} 0 & I_{h_1} \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ I_{w_1} & 0 \end{bmatrix}$
MB_2	Lower left	$\begin{bmatrix} 0 & I_{h_2} \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & I_{w_2} \\ 0 & 0 \end{bmatrix}$
MB_3	Upper right	$\begin{bmatrix} 0 & 0 \\ I_{h_3} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ I_{w_3} & 0 \end{bmatrix}$
MB_4	Upper left	$\begin{bmatrix} 0 & 0 \\ I_{h_4} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & I_{w_4} \\ 0 & 0 \end{bmatrix}$

Table 1: S_{i1} and S_{i2} matrices

Graphical representation of four neighbouring macroblocks (MBs) and motion vectors are shown in Figure 3. DCT coefficients in inter-coded frames can be extracted from an intra-coded frame. The computation of the DCT coefficients of a new arbitrary position image block from the DCT coefficients of the four original neighbouring blocks takes the form of pre-multiplying and post-multiplying of those blocks with appropriate matrices (S_{i1} and S_{i2}) [6]. Equation 3 describes this process of DCT coefficients evaluation of a motion-compensated MB. There are four possible locations of the sub-block of interest (with reference to MB_i): upper-left, upper-right, lower-left and lower-right. These locations define S_{i1} and S_{i2} matrices as tabulated in Table 1. Parameters h_i and w_i are the height and width of the overlap of MB_{pre} with MB_i . For a particular MB, these two parameters can be evaluated from its motion vector (u, v). Therefore, DCT can be evaluated for the predicted MB using Equation (3). DCT coefficients of the error term are readily available for MPEG-2 compressed video. Finally, DCT coefficients of the target MB are calculated by adding the DCT coefficients of the predicted MB and the DCT coefficients of the error signal.

3. PROPOSED VIDEO EDITING SYSTEM

Normally, motion estimation process bares the 60-70% of computational complexity of the MPEG video encoder. Therefore, in order to keep the computational complexity of the video editing process at a reasonable level, it is desirable to reuse the motion vectors as much as possible without re-computing them. Since we have extracted all DCT coefficients we can use the existing motion vectors by re-calculating the corresponding DFD signals. Therefore, we can produce fading operations either by intra-coding or motion compensating all MBs in P and B frames. All MBs in I-frames are intra-coded with new DCT coefficients, which are evaluated by Equation (4). Assume all MBs are intra-coded in P and B frames. Then fade-in operation can be applied to each MB separately as explained in Equation (4).

$$S_{n,m}(k_1, k_2) = \begin{cases} \left[1 - \left(\frac{n-L_1}{F} \right) \right] C + \left(\frac{n-L_1}{F} \right) G_{n,m}(0,0) & k_1, k_2 = 0 \\ \left(\frac{n-L_1}{F} \right) G_{n,m}(k_1, k_2) & \text{else} \end{cases} \quad (4)$$

where, $S_{n,m}(k_1, k_2)$ is DCT coefficients of $s_{n,m}(x, y)$, $G_{n,m}(k_1, k_2)$ is DCT coefficients of $g_{n,m}(x, y)$, m is MB number, k_1, k_2 are location in the MB ($k_1, k_2 = 0:7$). Similar arguments can be followed for fade-out as well.

If all MBs in every frame are intra-coded then, the number of bits to code them must be increased. This can be done in variable bit rate (VBR) system since number of bits is not limited. However, in constant bit rate (CBR) system number of bits is limited and hence the quantisation step size will be increased to maintain the same channel bit rate in this system. This will degrade the performance since we are not exploiting any correlation between frames. In this paper we consider a CBR system, which is widely used in practical systems. Therefore we cannot intra-code all MBs during a fading operation as picture quality will be degraded.

Now we assume that all MBs in P and B frames are motion compensated with existing motion vectors. But, these motion vectors are not optimised for the faded sequence. Thus, new DFD signal is comparatively large with respect to the old DFD signal. Therefore, if inter-coded frames are used continuously, picture quality will be degraded (especially high frequency components) as same quantisation matrices are used to quantise new DFD signals. Therefore, in this proposed scheme we use combination of intra-coded and motion compensated MBs. Thus we use existing motion vectors for all B-frames but all MBs in P-frames are intra-coded. However in B-frames, if new DFD signals are greater than a pre-determined threshold, those MBs are intra-coded. Our experiments show that this scheme produces the best picture quality compared to coding all P and B-frames as intra-coded or P and B-frames as motion compensated with existing motion vectors.

3.1 Implementation of the Proposed Video Editing System

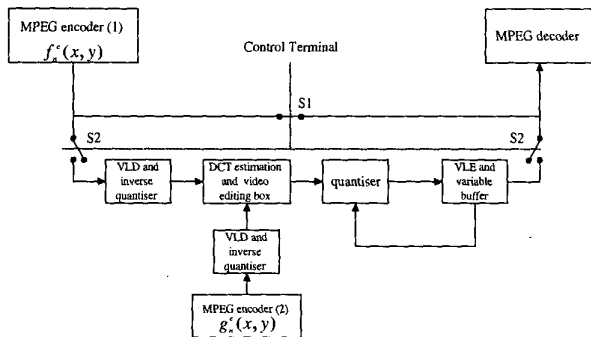


Figure 4: Proposed compressed domain video editor

Figure 4 shows the proposed compressed domain video editor. Two MPEG compressed video streams are applied to the editor and these two streams are partially decoded with variable length decoder (VLD) and the inverse quantiser. With this data, we estimate DCT coefficients for each frame as explained in section 2.2. Then these coefficients and motion vectors are applied to the video editing box to process the parameters as explained previously. Finally, all MBs are quantised to suit the channel bit rate. The quantisation is achieved through the feedback loop from the buffer as in normal MPEG-2 video encoder. When user needs to switch the video into a different sequence through any special effects operation, S2 switch will be closed and S1 switch will be opened through control terminal. The length of these fading operations can be controlled by the user, typically between 30-80 frames. There are twelve locations, which fade-in can be terminated within a GOP (12,3) structure. It should be noted that the last frame in the fade-in operation is an unaltered picture from the second sequence, $g_n(x, y)$. Therefore, if the last frame of the fade-in process is an I-picture or a P-picture (originally), then no further processing is required for following frames to synchronise with the sequence. Depending on whether the last frame is a B1-picture or B2-picture all MBs are needed to be intra-coded in one or two future pictures respectively. When the last frame is a B1-picture (2,5,8,11), then all MBs are needed to be intra-coded in the next two frames since motion vectors for these two frames are incorrect. When the last frame is a B2-picture (3,6,9,12), then all MBs should be intra-coded in the following frame. Similar arguments can be followed for fade-out as well.

4. RESULTS

Here, we considered some simulations of fade-in and fade-out video special effects editing using the proposed scheme. We used MPEG-2 video streams as input to the proposed compressed domain video editor and we considered a CBR system. We compared our results with the conventional scheme (Figure 2) and DC-only scheme [5]. PSNR of the luminance signal is considered to measure the objective quality of the edited video using these three techniques. We considered the signal $s_n(x, y)$ (output at the uncompressed video editor – Figure 1) as the reference signal for all PSNR calculations. Figure 5 shows the comparison for a fade-in special effect editing in compressed domain. Figure 6 shows the same for fade-out special effect editing. Results show that PSNR of the DC-only scheme [5] is very low. PSNR of the DC-only scheme is decreasing, until it finds the next I-frame. Even at I-frames PSNR is well below the expected value. But, PSNR of the proposed scheme closely follows the PSNR of the conventional scheme. PSNR of the DC-only scheme is heavily dependent on the solid colour and the nature of the MPEG-2 video sequences. The proposed scheme is independent of these parameters. Figure 5 also shows that

PSNR of the DC-only scheme is very low even after the fade-in operation, until it finds the next I-frame. This is due to accumulated error in the last frame of the fade-in sequence and the failure to provide a solution to end point synchronisation. This problem cannot be seen in Figure 8 since fade-out ends exactly at the last frame of a GOP. In the proposed scheme, we made allowances for various end points in the MPEG sequence (i.e. I, P or B frames). Figure 7 shows the subjective quality of the 70th frame of fade-in operation. As shown, subjective quality of the DC-only method is poor compared to the other two schemes.

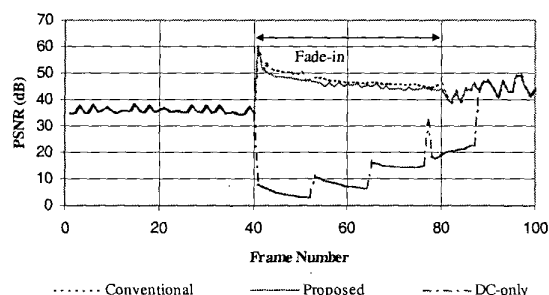


Figure 5: Performance comparison for fade-in production ($F = 40$, $C = 20$)

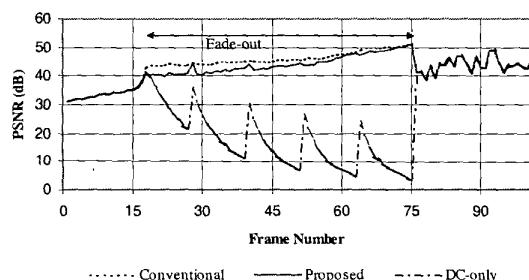


Figure 6: Performance comparison for fade-out production ($F = 56$, $C = 140$)



Figure 7: 70th frame of the fade-in operation (from left, conventional, proposed and DC-only)

Conventional scheme is computationally very expensive compared to the proposed scheme. Although DC-only method is less computationally expensive, the quality of the edited video is very poor. We tested the proposed scheme with different fading rates and different MPEG-2 video bit streams and observed similar set of results. Some of these results are

tabulated in table 2. Therefore, these special effects editing can be done in compressed domain itself without full frame decompression and motion re-estimation.

Description	F	Average PSNR (dB)		
		Conventional	Proposed	DC-only
Fade-out	40	47.8	46.4	10.4
Fade-out	50	47.4	46.6	11.2
Fade-in	50	45.8	44.9	9.9
Fade-in	30	48.4	47.4	10.5
Fade-in	30	48.2	47.3	10.1

Table 2: Summarised results (F- Length of the fading sequence)

5. CONCLUSIONS

In this paper we focused on developing fade-out and fade-in video editing operations in MPEG-2 compressed domain without full frame decompression and motion re-estimation. We estimated DCT coefficients for all frames and used these DCT coefficients together with the existing motion vectors to produce these special effects in compressed domain. Results show that the subjective and objective quality of the edited video closely follows the quality of the edited video with the conventional method. Unlike the conventional scheme, proposed scheme is computationally inexpensive and makes it possible for real time implementations. Future work is required to extend this work for dissolve and wipe editing in compressed domain.

ACKNOWLEDGEMENTS

First author would like to express his gratitude and sincere appreciation to the University of Bristol and CVCP for providing financial support for this work.

REFERENCES

1. Sikora, T., "MPEG Digital Video-Coding Standards", IEEE Signal Processing magazine, pp. 82-100, September 1997.
2. ITU-T, "Draft H.263", May 1996.
3. Smith, B.C., Rowe, L.A., "A New Family of Algorithms for Manipulating Compressed Images", IEEE Computer Graphics and Applications, pp. 34-42, September 1993.
4. Wallace, G.K., "The JPEG still picture Compression Standard", Communications of the ACM, Vol. 34, pp. 30-44, April 1991.
5. Shen, B., "Fast Fade-out Operation on MPEG Video", ICIP, CD Paper Number - MP10.01, 1998.
6. Chang, S.F., Messerschmitt, D.G., "Manipulation and compositing of MC-DCT compressed video", IEEE Journal of Selected Areas in Communications, vol.13, pp.1-11, January 1995.